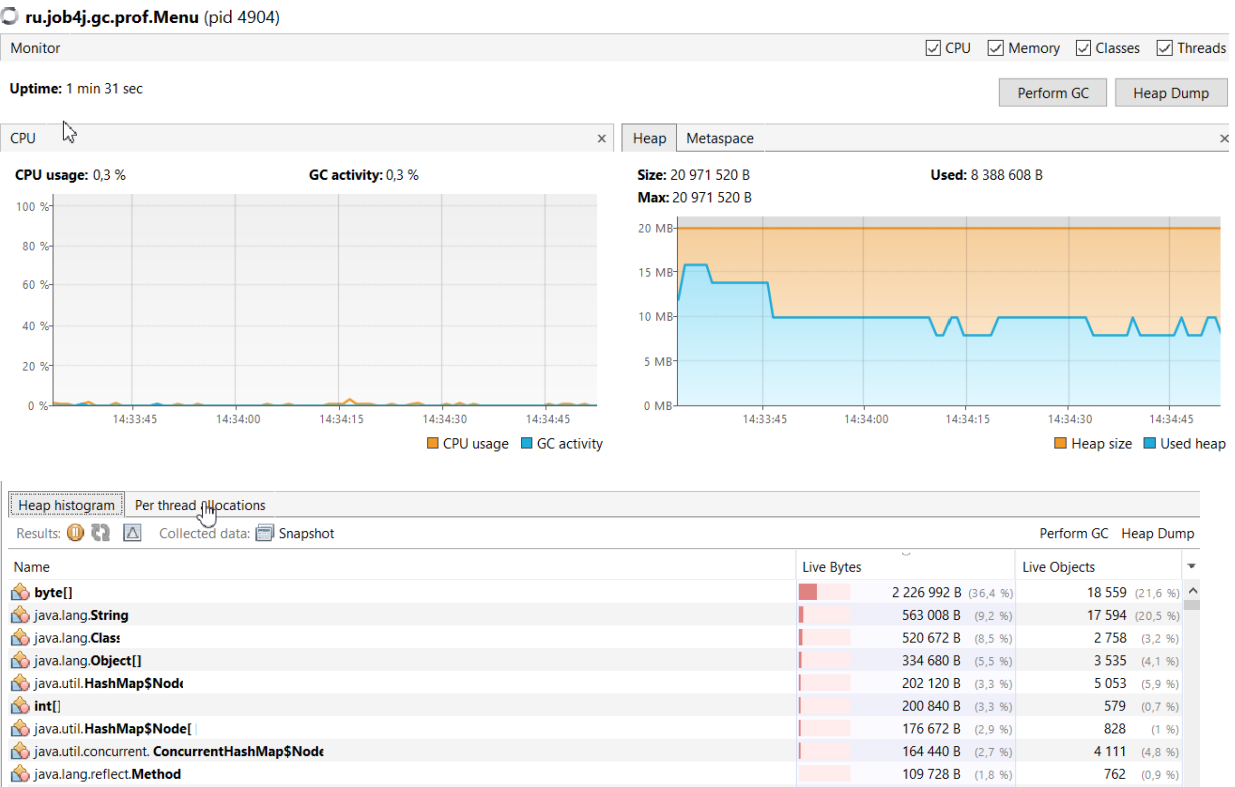


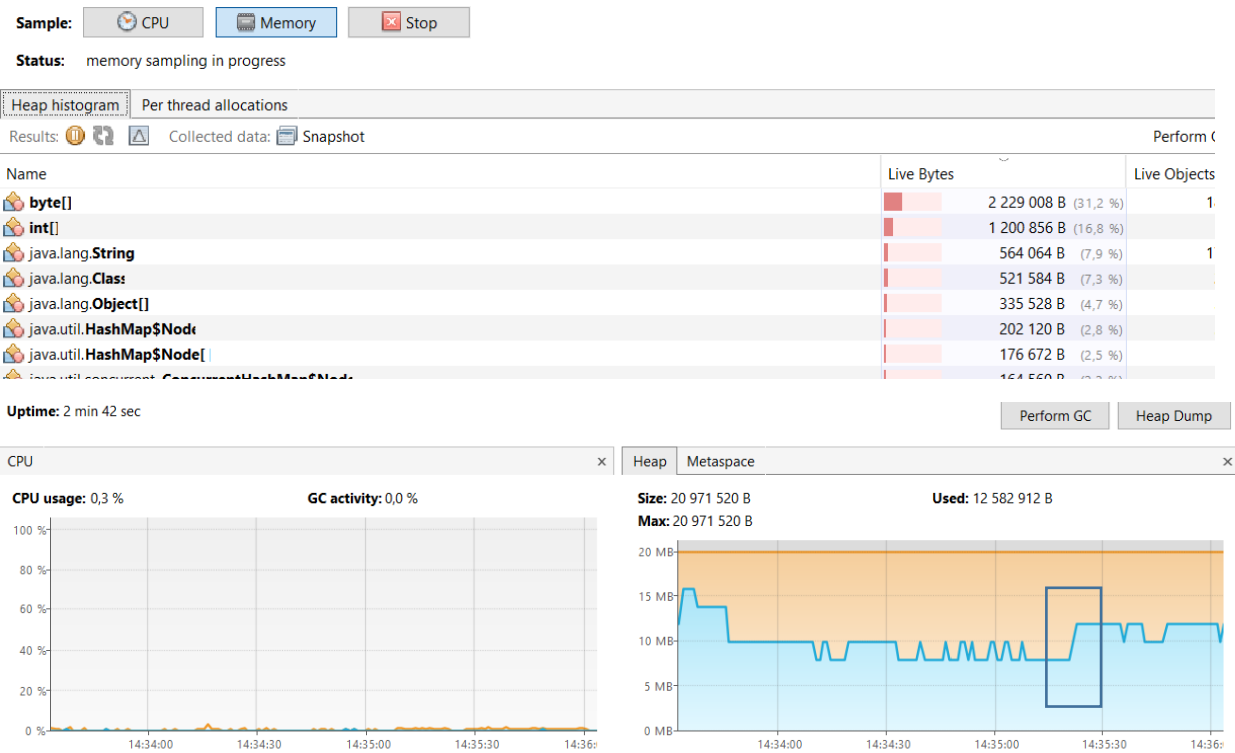
ZGC

график хипа при запуске программы не выдает равномерных колебаний как в Parallel или G1.

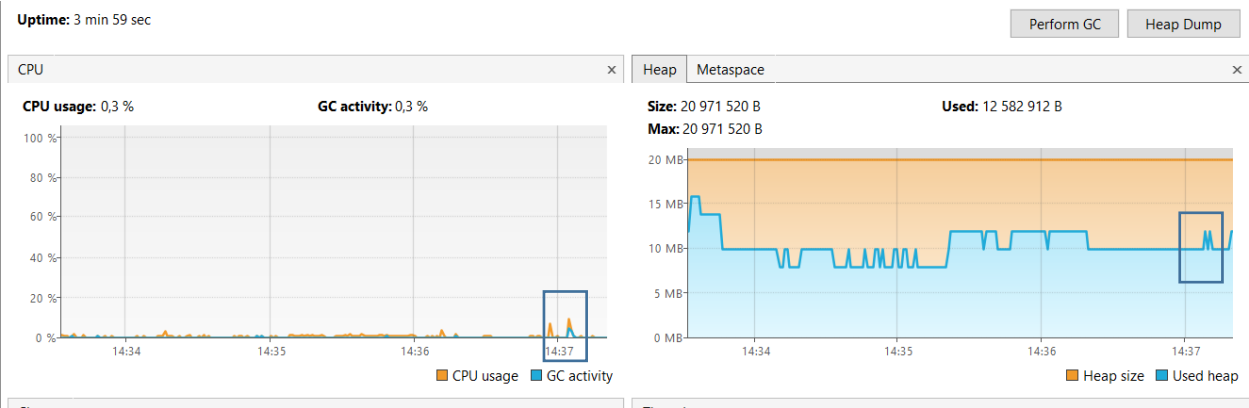
прямых участков стало больше.как я понял периодически проходят супер малые сборки(мелкие , в логах не указано время остановки, тк работает параллельно и не требует остановок. Мало заметна ,но все же видна активность GC на графике CPU – синий цвет



Насоздание массива стандартно выделилась память, значение используемой памяти выросло



Все также быстра провелаь сортировка слиянием ~ 14 мс



Heap histogram Per thread allocations

Results: Collected data: Snapshot

Name	Live Bytes	Li
byte[]	2 231 936 B (31,3 %)	
int[]	1 200 856 B (16,8 %)	
java.lang.String	566 240 B (7,9 %)	
java.lang.Class	521 208 B (7,3 %)	
java.lang.Object[]	335 104 B (4,7 %)	

На сортировку вставками потребовалось: 8 сек времени, дополнительно 1000000байт на клон массива и успела произойти сборка мусора

Sample: CPU Memory Stop

Status: memory sampling in progress

Heap histogram Per thread allocations

Results: Collected data: Snapshot

Perform GC Heap Dump

Name	Live Bytes	Live Objects
byte[]	2 232 248 B (27,4 %)	18 669 (21,7 %)
int[]	2 200 872 B (27 %)	581 (0,7 %)
java.lang.String	566 528 B (7 %)	17 704 (20,6 %)
java.lang.Class	521 208 B (6,4 %)	2 761 (3,2 %)
java.lang.Object[]	335 104 B (4,1 %)	3 546 (4,1 %)
java.util.HashMap\$Node	202 120 B (2,5 %)	5 053 (5,9 %)
java.util.HashMap\$Node[]	176 672 B (2,2 %)	828 (1 %)
java.util.concurrent.ConcurrentHashMap\$Node	164 600 B (2 %)	4 115 (4,8 %)

ru.job4j.gc.prof.Menu (pid 4904)

Sampler Settings

Sample: CPU Memory Stop

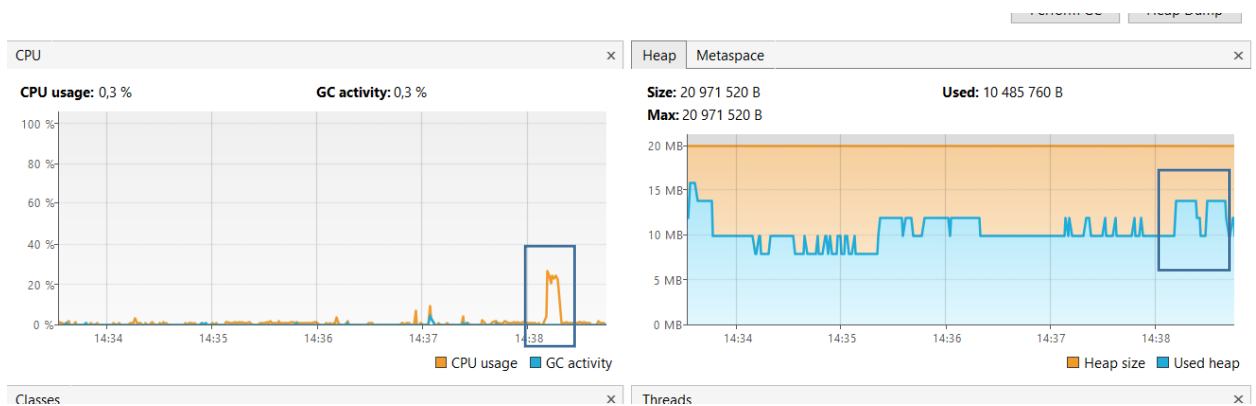
Status: memory sampling in progress

Heap histogram Per thread allocations

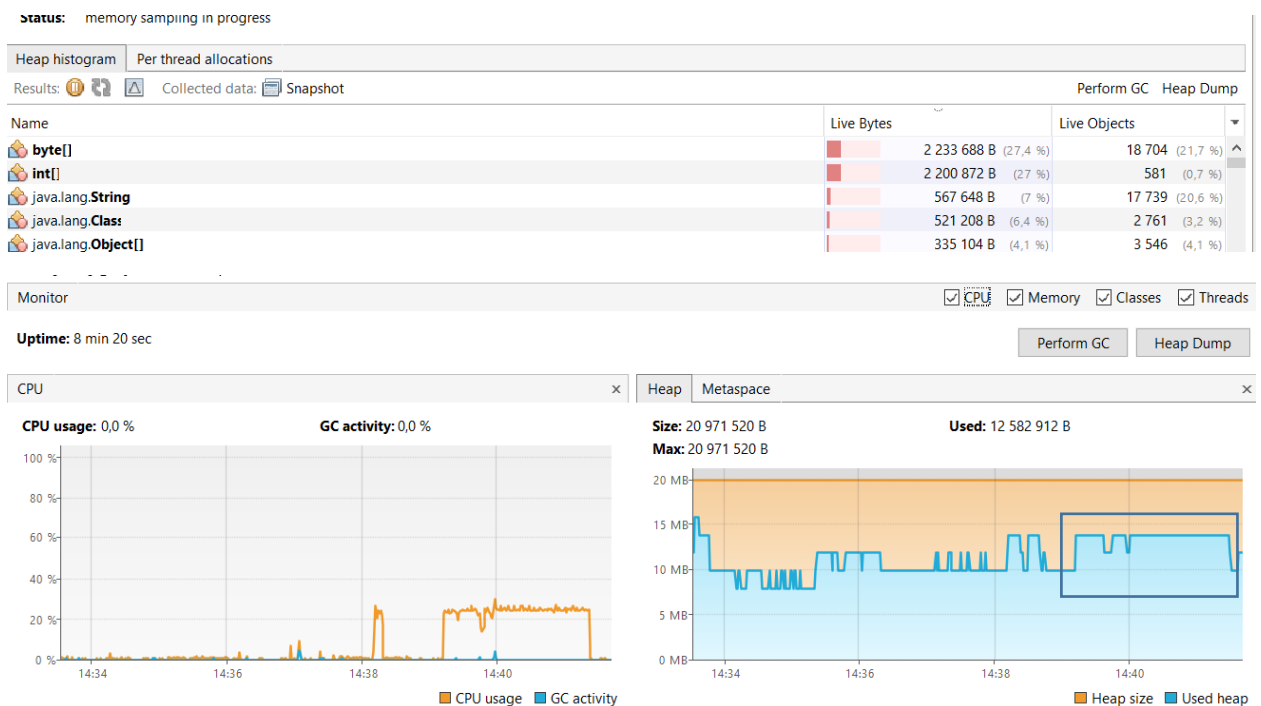
Results: Collected data: Snapshot

Perform GC Heap Dump

Name	Live Bytes	Live Objects
byte[]	2 232 944 B (31,3 %)	18 686 (21,7 %)
int[]	1 200 856 B (16,8 %)	580 (0,7 %)
java.lang.String	567 072 B (7,9 %)	17 721 (20,6 %)
java.lang.Class	521 208 B (7,3 %)	2 761 (3,2 %)
java.lang.Object[]	335 104 B (4,7 %)	3 546 (4,1 %)
java.util.HashMap\$Node	202 120 B (2,8 %)	5 053 (5,9 %)



Сортировка пузырьком заняла 2 минуты, в процесс также были небольшие сборки мусора



По времени данный GC показал самую медленную работу среди всех сборщиков, однако благодаря механизму параллелизма и многопоточности программа работала без остановок gc

